Reproducible Experiments on Adaptive Discriminative Region Discovery for Scene Recognition

Zhengyu Zhao
Radboud University, Netherlands
z.zhao@cs.ru.nl

Zhuoran Liu
Radboud University, Netherlands
z.liu@cs.ru.nl

Martha Larson
Radboud University and TU Delft, Netherlands
m.larson@cs.ru.nl

Ahmet Iscen*
Google Research
iscen@google.com

Naoko Nitta*
Osaka University, Japan
naoko@comm.eng.osaka-u.ac.jp

ABSTRACT

This companion paper supports the replication of scene image recognition experiments using Adaptive Discriminative Region Discovery (Adi-Red), an approach presented at ACM Multimedia 2018. We provide a set of artifacts that allow the replication of the experiments using a Python implementation. All the experiments are covered in a single shell script, which requires the installation of an environment, following our instructions, or using ReproZip. The data sets (images and labels) are automatically downloaded, and the train-test splits used in the experiments are created. The first experiment is from the original paper, and the second supports exploration of the resolution of the scale-specific input image, an interesting additional parameter. For both experiments, five other parameters can be adjusted: the threshold used to select the number of discriminative patches, the number of scales used, the type of patch selection (Adi-Red, dense or random), the architecture and pre-training data set of the pre-trained CNN feature extractor. The final output includes four tables (original Table 1, Table 2 and Table 4, and a table for the resolution experiment) and two plots (original Figure 3 and Figure 4).

KEYWORDS

Scene recognition; adaptive discriminative region discovery; multi-scale feature aggregation; reproducibility

ACM Reference Format:
Zhengyu Zhao, Zhuoran Liu, Martha Larson, Ahmet Iscen*, and Naoko Nitta*. 2019. Reproducible Experiments on Adaptive Discriminative Region Discovery for Scene Recognition. In *Proceedings of the 27th ACM International Conference on Multimedia (MM '19), October 21–25, 2019, Nice, France.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3343031.3351169

* Reviewers for this reproducibility paper.

MM '19, October 21–25, 2019, Nice, France

ACM ISBN 978-1-4503-6889-6/19/10.
https://doi.org/10.1145/3343031.3351169

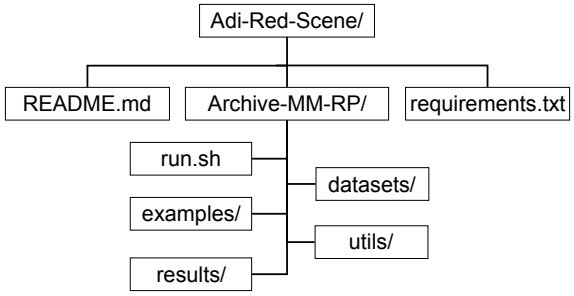

Figure 1: File structure of the replication artifacts

1 INTRODUCTION

1.1 Summary of Adi-red

In the original paper [4], we proposed Adi-Red, a novel adaptive discriminative region discovery approach for scene recognition. Adi-Red exploits local discriminative information from a CNN model that has been pre-trained on a large scene-centric dataset (Places [5]). Along with a simple threshold-based patch selection strategy, the number of local discriminative regions is allowed to vary adaptively for each image. A particularly useful advantage of Adi-Red is that it reduces the computational effort needed to select discriminative patches over existing approaches.

1.2 Functionality of the replication release

Our replication release provides a complete implementation of the experiments in the original paper, plus an additional resolution experiment (described in 2.6). Moreover, it is possible to adjust the parameters in this release. The artifacts that are relevant to this paper have been placed on a publicly accessible GitHub repository[1], which contains a detailed `README` file and required scripts for running experiments. The file structure of the artifacts is shown in Figure 1.

[1] https://github.com/ZhengyuZhao/Adi-Red-Scene

Table 1: Notations of Parameters

Parameter	Value in original paper	Description
T	T=150 for Local_1 T=100 for Local_2	A threshold on the local maxima of Dis-Map. Locations with values higher than the threshold are selected as centers of the discriminative regions.
Selection_type	random/dense/Adi-Red	Type of patch selection
Total_scales	3 (Global/Local_1/Local_2)	The total number of scales used for multi-scale patch feature aggregation.
Arch	AlexNet/ResNet18/ResNet50	The architecture of the pre-trained CNN feature extractor
Pre_trained_data	PL/PL/IN	Different CNN feature extractors pre-trained by different datasets used at each scale. PL denotes Places and IN denotes ImageNet.
resolution	original	The original resolution of input image at each scale before being resized into $224 \times 224/448 \times 448/896 \times 896$ respectively. Low-resolution version will instead use pre-resized image with the size of 224×224.
Win_size & Stride	3x3&1	The sliding window is applied for searching local maxima of the Dis-Map with a given stride. We select the center values in each window that are equal or greater than all of their surrounding pixels as the local maxima.

2 EXPERIMENTS

In this section, we describe the experiments in details, and also discuss the differences between the replication and original results.

2.1 Environment Installation

The following requirements are needed for the replication.

(1) Python3 (tested with Python 3.7.2 on Ubuntu 16.04.6 LTS), required libraries can be found in the requirements.txt from the GitHub repository and installed by running:
`pip3 install -r requirements.txt`
(2) PyTorch deep learning framework (tested with version 1.0.1) and torchvision (tested with version 0.2.2)
(3) CUDA driver and corresponding cuDNN package if using GPU (tested using Nvidia P100 with CUDA 8.0 and cuDNN 7.1.2)

To install PyTorch and CUDA packages, please refer to their official websites for versions that are compatible.

Alternatively, the open source tool ReproZip [2] can be used. After downloading the ReproZip package[2], you can automatically set up the same environment following the instructions below:

```
reprounzip docker setup Adi-Red-Scene.rpz Adi-Red-Experiment
reprounzip docker run Adi-Red-Experiment
```

About 300G disk space is needed to store the datasets and results.

2.2 Process Description

Here we describe the scripts that are called in the main shell script `run.sh` when running our experiments. Please refer to the `README` file for detailed usage and a description of the adjustable parameters.

- `prepare.sh` is run to create structured folders and download the datasets.
- `data_clean.py` is run to assign the data into required train/test splits and conduct label-level pre-processing operations.
- `dis_map_generation.py` describes how to generate the discriminative map (Dis-Map) for each scene image in the dataset (Places/SUN397) by using a discriminative discovery network (DisNet), which was pre-trained on the large-scale Places365-Standard dataset.
- `adaptive_region_selection.py` is run to determine the image locations where the discriminative patches will be cropped at each of two local scales.
- `intra_scale_feature_extraction.py` is run to extract deep features from each image at different scales.
- `svm_classification.py` is run to obtain the final image representation based on sequential operations including feature normalization, concatenation and scaling, and implement the classification with the SVMs.
- `plot.py` is run to automatically generate the tables and plots corresponding to those in the original paper.

In addition, a simple example test in `demo.py` that covers the key elements of Adi-Red can be run to output the final image representation for the example image, and a heatmap that expresses local discriminativeness, which is actually similar to Figure. 5 in the original paper.

2.3 Datasets

The examples in the replication release use the same datasets as is in the original paper, and also the same training and test splits. Here, we describe the two datasets in turn.

SUN397 is a widely used scene-centric database, which includes 397 scene categories with at least 100 images for each category. We follow the standard evaluation protocol reported in the original paper [3], which uses 50 images for training and 50 images for testing per category. The average classification accuracy over the provided ten train/test splits is reported.

Places365-Standard dataset [5] was created by selecting 365 categories that contain more than 4000 images (totally 1.8 million training image) from the large-scale Places database. We only conduct experiments on its official validation set, which includes 100 images per category. For each category, we select the first half of images (in alphabetical order of their file names) as training data and the rest for testing.

The data (datasets and labels) will be automatically downloaded during the replication.

[2]See the `README` file in our GitHub repository for the link

Table 2: Scene recognition with two different resolutions of the original image. (Pre-training datasets are reported for the three scales: global/coarse local/fine local; PL= Places and IN=ImageNet)

# of scales (global+local)	Pre-training dataset	Low resolution	Adi-Red
1+1	PL/-/IN	39.84	40.76
1+1	PL/PL/-	41.24	42.16
1+2	PL/PL/IN	41.60	42.57

2.4 Parameter Settings

The parameters that can be adjusted in the experiments are listed in Table 1, along with their descriptions. Note that we allow the adjustment of the parameter Win_size and Stride, but did not report any comparison in the paper.

2.5 Main Experiment

In our replication release, we provide a pipeline that makes it possible to reproduce the same experiments carried out in our original paper. We expected the output to be very similar to what was reported in the original paper, however, it might not be identical. In general, since we leave out the operation of saving the Dis-Map as JPEG-compressed image before patch selection and instead implement Adi-Red as a single pipeline, more accurate discriminative information will be captured than that from the compressed Dis-Map, where fewer local maxima can be found due to smoothing. By eliminating such unnecessary loss of discriminative information, the results of Adi-Red in the replication will be consistently better than those in the original paper. This explanation holds except for the case with ResNet50, as can be seen in Table 3. We conjecture that it is because that this deep architecture has already captured good discriminative information by itself, so that the discriminative regions with relatively smaller local maxima introduced by Adi-Red will somehow harm it.

Moreover, the results may also be affected by the following factors:

(1) For the replication release, we re-implemented the original experiments on patch selection and SVM classification in Python. They were carried out in MATLAB in the original paper. We chose Python in order to make the replication release implemented with only a single language, in a continuous pipeline. This transfer might cause some differences due to the change of certain libraries.
(2) The two types of patch selection (dense and random) implemented in the replication release for comparison with Adi-Red make a random choice of which patches to crop. This randomness obviously leads to small differences in the results (as compared to Table 1 of the original paper) across different sessions of replication.

Since the discrepancies are small as shown in Table 3 and Table 4, we conclude that the original high-level contributions of our paper stand after the replication: namely, "We demonstrate the strength of using an adaptive number of regions per image to extract features capturing local-level information." and "Adi-Red improves efficiency by eliminating the need for both feature extraction from a large number of local patches and clustering operations, which are common in current methods." (see Introduction of the original paper). Because these conclusions still clearly hold, we consider the replication to be successful.

Table 3: Differences between the original and replication results on SUN397 in terms of Adi-Red versus single-scale baseline based on three different CNN architectures

Networks	original		replication	
	Baseline	Adi-Red	Baseline	Adi-Red
AlexNet	54.17	61.01	53.87	61.51
ResNet18	66.96	70.58	66.99	70.88
ResNet50	71.38	73.59	71.14	73.32

Table 4: Differences between the original and replication results on SUN397 in terms of different patch selection types

	random	dense	Adi-Red
original	56.90	60.49	61.01
replication	57.53	60.64	61.51

However, we also point out that the small differences between the original implementation and the replication release meant that the ResNet50-based Adi-Red cannot be claimed to outperform the state-of-the-art approach [1] (not included in table). In the original paper, we mentioned that it outperformed the state of the art, and with this replication we revise that claim and conclude that Adi-Red performs on par with the state of the art. The fundamental advantage of Adi-Red over [1] of course remains: Adi-Red use fewer scales and fewer patches per image.

For different selection types, as shown in Table 4, we see the underlying conclusion that Adi-Red outperforms other two types of patch selection with fewer patches still stands.

2.6 Resolution Experiment

In some use cases, the input image with original resolution might not be available, so that it is interesting to explore the impact of low-resolution input image (first downsampled 224×224 at each scale) on Adi-Red.

The results are shown in Table 2. We can observe that although lower resolution results in some decrease of performance, but still, Adi-Red can outperform the single-scale baseline (38.53) by a large margin.

2.7 Output

The final output of our replication release consists of four tables (Table 1, Table 2 and Table 4 in the original paper, and another table for the resolution experiment) showing the average accuracy over scene categories with different adjustable parameters, a graph showing the average number of discriminative patches used by

Adi-Red per category (original Figure 3), and a graph showing classification accuracy as a function of the varying threshold (original Figure 4).

3 REVIEWING PROCESS

The provided package contains a main script (`run.sh`) which calls every other function needed to reproduce the experiments. Note that the authors provide an implementation that is different from the original paper implementation (see Section 2.5), therefore some functions either exited with exceptions or produced unexpected results early in the reviewing process. However, the authors quickly responded to the issues brought up by the reviewers, and provided necessary corrections to enable reproduction of the experiments.

The main script (`run.sh`) sequentially runs every script necessary to reproduce all the tables. However, it is also possible to run some of these scripts in parallel to reduce the execution time dramatically. For example, scripts for CNN feature extraction for different tables and figures can be run in parallel, followed by scripts for SVM classification.

In conclusion, two reviewers and the authors worked together for this reproduction paper. Minor corrections to the code were made during the process. Communication between the authors and reviewers was efficient and timely. The reviewers were able to reproduce expected results after the process.

4 CONCLUSION

In this paper, we have documented a successful replication of the original Adi-Red paper. We have discovered some small, explainable discrepancies between the original implementation and our replication release. These do not change the main claims in the original paper that Adi-Red makes use of discriminative patch selection to achieve state-of-the-art performance, given the use of the proper number of scales, and an optimized setting of the number of patches per image.

ACKNOWLEDGMENTS

This work was carried out on the Dutch national e-infrastructure with the support of SURF Cooperative.

REFERENCES

[1] Xiaojuan Cheng, Jiwen Lu, Jianjiang Feng, Bo Yuan, and Jie Zhou. 2018. Scene recognition with objectness. *Pattern Recognition* 74 (2018), 474–487.

[2] Fernando Chirigati, Rémi Rampin, Dennis Shasha, and Juliana Freire. 2016. Reprozip: Computational reproducibility with ease. In *Proceedings of the 2016 International Conference on Management of Data*. ACM, 2085–2088.

[3] Jianxiong Xiao, Krista A. Ehinger, James Hays, Antonio Torralba, and Aude Oliva. 2016. Sun database: Exploring a large collection of scene categories. *International Journal of Computer Vision (IJCV)* 119 (2016), 3–22.

[4] Zhengyu Zhao and Martha Larson. 2018. From Volcano to Toyshop: Adaptive Discriminative Region Discovery for Scene Recognition. In *2018 ACM Multimedia Conference on Multimedia Conference*. ACM, 1760–1768.

[5] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. 2018. Places: A 10 million image database for scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* 40 (2018), 1452–1464.